시계열 파운데이션 모델을 활용한
실시간 PC 시스템 이상탐지 파이프라인 구축

Team 7

문세희 윤사무엘 이서진 이재찬 하주영

2025.10.16

DE Team Project Proposal

Problem Definition

- **실시간 PC 시스템 이상탐지**

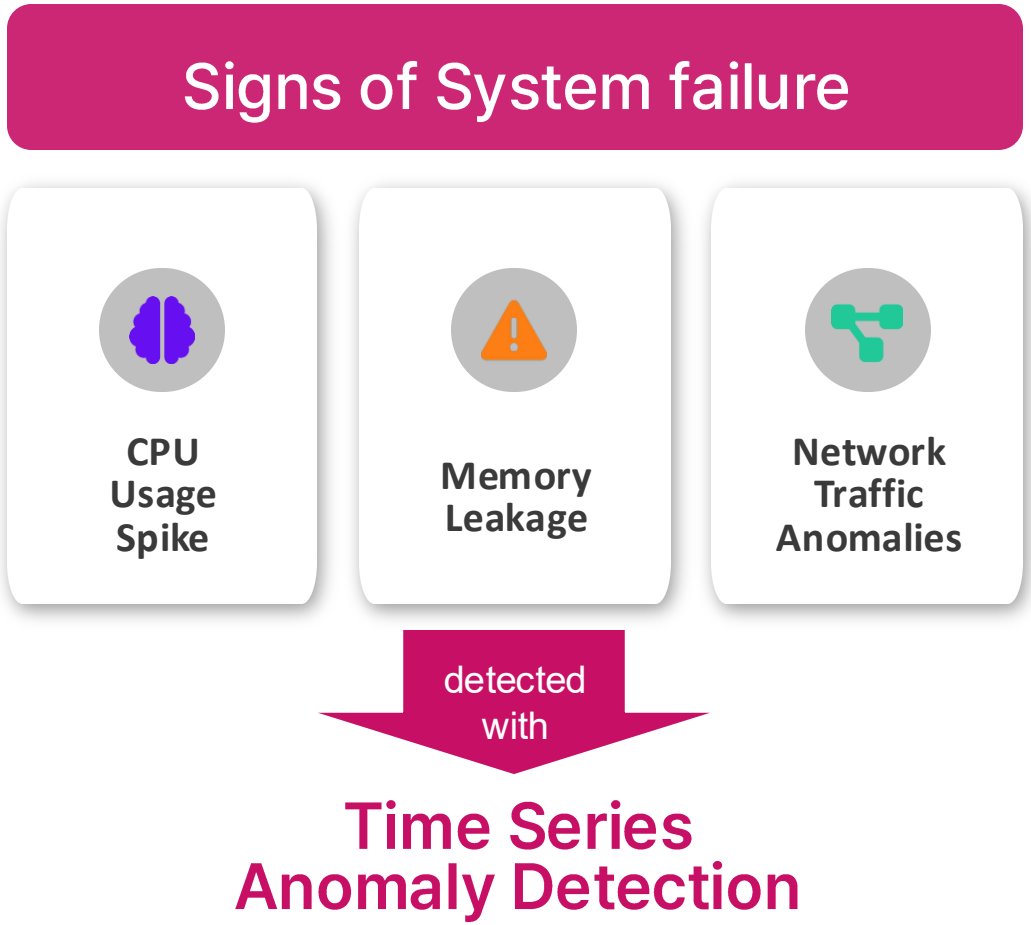
- **시스템 장애와 외부 공격의 징후**

  ✓개인 PC부터 기업 서버에 이르기까지, 시스템 장애나 외부 공격은 CPU 사용량 급증, 메모리 누수, 네트워크 트래픽 이상 등의 형태로 나타남

  ✓이러한 미세한 이상 징후를 조기에 자동으로 탐지하여 시스템 안정성을 확보하는 것은 매우 중요함

- **IT 인프라 운영의 핵심 과제**

  ✓본 프로젝트는 안정적인 IT 인프라 운영의 핵심인 “시스템 모니터링” 문제를 직접 해결하는 것으로, 산업 현장에서 중요하게 다루어지는 주제임



Problem Definition

Case 1



작업물 손실 방지

✕ 메모리 과부하로 프로그램이 중단되어
작업중이던 자료가 날아간 상황 (ex. PPT)

☑ 점진적 메모리 누수를 탐지하여
작업물의 손실을 미연에 방지함

Case 2



주기적 백업 실패 방지

✕ 주기적으로 작동해야 하는 사진 백업
프로그램이 특정 문제로 작동되지 않음

☑ 주기적 패턴이 소멸되는 현상을 감지하여
자동화된 백업 실패를 조기에 경보함

Case 3



보안 위협 조기 경고

✕ 해커의 공격으로 인해 PC 내부의 데이터가
네트워크를 통해 대량으로 유출됨

☑ CPU 사용량에 비해 네트워크 트래픽이
과도하게 증가하는 '비정상적인 상관관계'
탐지, 잠재적 보안 위협 식별

Objectives

▪ End-to-End 데이터 엔지니어링 파이프라인 구축

- **데이터 수집부터 사용자 알림까지**
  - ✓ 오픈소스 데이터셋이 아닌, 우리 팀이 직접 사용하는 PC에서 생성되는 실시간 스트리밍 데이터 사용 (CPU, 메모리, 네트워크)
  - ✓ 데이터 발생 지점부터 이상 탐지 결과가 대시보드에 시각화되고 사용자에게 경보가 발송되는 순간까지,
    데이터 흐름의 전 과정을 직접 설계하고 구현
- **시계열 이상탐지 파운데이션 모델**
  - ✓ 고성능 시계열 파운데이션 모델을 시계열 이상탐지에 사용하여 문제 해결에 필요한 모델링 직접 수행

Why TSFM?

▪ 시계열 파운데이션 모델이 본 프로젝트에 적합한 이유

- **맥락 이해 능력**

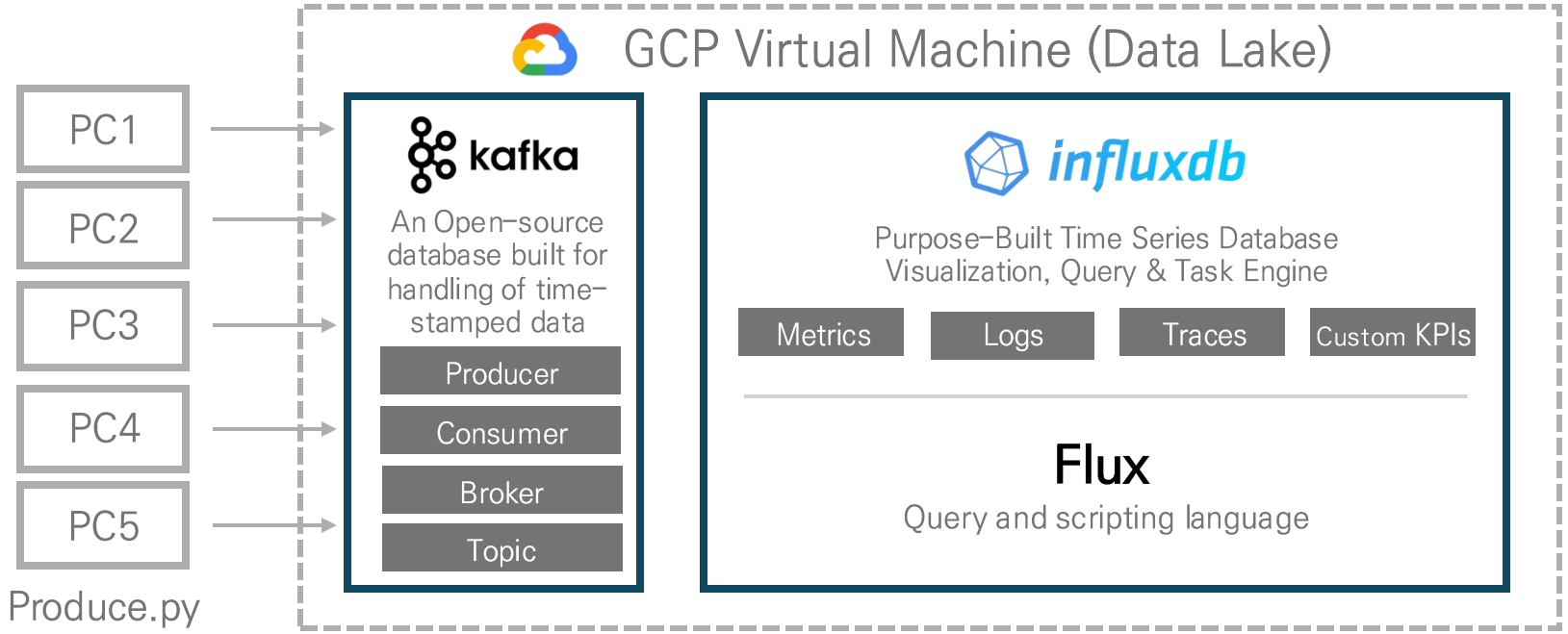
  ✓ 점진적으로 발생하는 contextual anomaly를 포착하고, 있어야 할 패턴 인식에 실패하거나 지표 간 상관관계가 깨지는 등 복합적인 이상 신호를 효과적으로 감지함

- **데이터 효율성**

  ✓ 데이터 수집의 한계라는 현실적 제약 속에서, 방대한 양의 데이터로 사전 학습된 모델이 타 도메인으로부터 얻은 지식 전이를 통해 소량의 데이터만으로도 높은 성능을 보장함

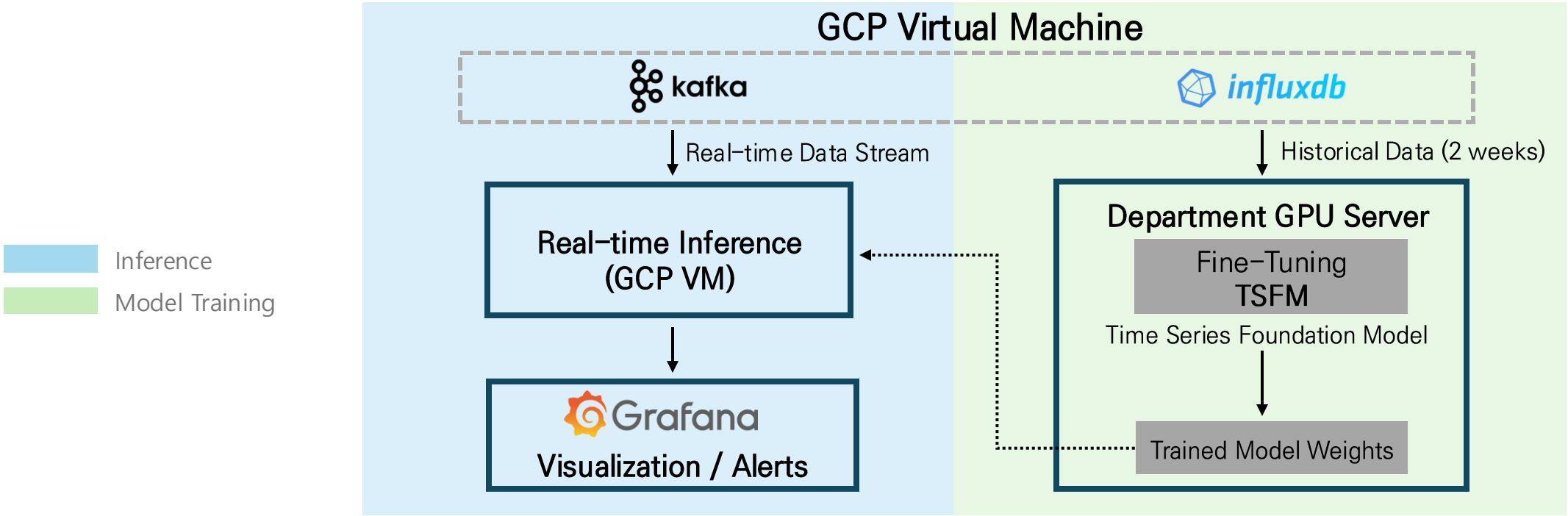
Model	Architecture	Univariate/Multivariate	Parameters	Data	Patch	Loss	Probabilistic
Tiny Time Mixers	Non-Transformer	Both	1M	1B	Yes	MSE	No
Timer-XL	Decoder	Both	Unknown	1B	Yes	MSE	No
Time-MoE	Decoder	Both	2.4B	309B	No	Huber	No
Toto	Decoder	Both	103M	1T	Yes	Neg Log-likelihood	Yes
TimeGPT	Encoder-Decoder	Univariate	Unknown	100B	No	Unknown	No
Chronos	Encoder-Decoder	Univariate	8M, 46M, 201M, 710M	84B	No	Cross Entropy	Yes
MOMENT	Encoder	Univariate	40M, 125M, 385M	1.13B	Yes	MSE	No
MOIRAI	Encoder	Both	14M, 91M, 311M	27B	Yes	Log-likelihood	Yes
AutoTimes	LLM	Both	GPT-2, OPT-350M, LLaMA-7B	Same as LLM	Yes	MSE	No
Timer	Decoder	Univariate	29M, 50M, 67M	28B	Yes	MSE	No
TimesFM	Decoder	Univariate	200M	100B	Yes	MSE	No
Lag-Llama	Decoder	Univariate	200M	352M	No	Neg Log-likelihood	Yes
LLMTime	LLM	Univariate	Llama 70B, GPT-3/4	Same as LLM	No	Unknown	Yes
Time-LLM	LLM	Both	Llama-1 70B, GPT-3	Same as LLM	Yes	MSE	No
FPT	LLM	Univariate	GPT-2: BERT, BEiT	Same as LLM	Yes	MSE	No

데이터 수집 & 저장 파이프라인



1. **데이터 수집 (Source)**
   - ✓ 팀원 로컬 PC 5대에서 producer.py를 통해 시스템 자원 데이터 수집
2. **데이터 전송 (Data Transmission)**
   - ✓ 수집된 데이터 스트림을 GCP VM에 구축된 Kafka로 전송
3. **실시간 처리 및 버퍼링 (Real-time processing & Buffering)**
   - ✓ 클라우드의 Kafka가 전송된 데이터를 받아 안정적으로 관리
4. **시계열 저장 (Time-series Storage & Query)**
   - ✓ Kafka에 대기하던 데이터를 InfluxDB 시계열 데이터베이스에 저장

모델 학습 & 실시간 추론 파이프라인



1. **학습 데이터셋 구축 (Training Dataset Preparation)**
   - ✓ InfluxDB에 2주간의 Historical Data를 학과 GPU 서버로 가져와 전처리 진행
2. **모델 학습 (Model Training)**
   - ✓ 학과 GPU를 활용하여 시계열 파운데이션 모델을 Fine-tuning
3. **실시간 데이터 스트리밍 (Real-time Data Streaming)**
   - ✓ GCP의 Kafka를 통해 새로운 시스템 데이터가 실시간 스트림 형태로 전달
4. **실시간 이상 탐지 (Real-time Anomaly Detection)**
   - ✓ 학습된 모델로 GCP VM 메모리에서 실시간 이상 탐지 수행

Data Collection & Pre-processing

▪ 안정적인 모델 학습을 위한 데이터 수집 및 전처리 전략

- **데이터 수집 및 구성**
  - ✓ 팀원 5명의 PC에서 2주간 '정상' 및 '이상' PC 사용 데이터를 수집 (매 5초마다 데이터 생성)
  - ✓ 사전에 정의된 이상 시나리오에 따라 인위적으로 이상 상황을 발생시켜 '이상' PC 데이터 확보 (이외에는 '정상'으로 간주)
- **데이터 스키마**
  - ✓ Tags: host_id, os
  - ✓ Fields: cpu_total_pct (float), mem_used_pct (float), net_in_bps (int), net_out_bps (int), missing_mask (int)
  - ✓ Time: ts(UTC)
- **데이터 전처리: 윈도잉/ 결측치 보간/ 스케일링**
  - ✓ 일정한 시퀀스 길이와 stride를 결정하여 슬라이딩 윈도우 방식으로 학습 데이터를 생성
  - ✓ 10초 이내 결측 시 Forward Fill 로 처리하고, 그 이외에는 N/A 에 해당하는 값 (ex.-999) 채움
  - ✓ 각 PC 별로 이질적인 데이터를 효과적으로 처리하고, Outlier에 강건하도록 PC별 Robust Scaler (IQR 기반) 적용

x_scaled = (x − median) / IQR

Model Training & Experiments

▪ 모델 학습 및 실험 방안

- **파운데이션 모델 학습**
  - ✓ MOMENT, Chronos 등 몇 가지 후보 모델 중 파일럿 실험을 통해 성능이 좋은 1가지 모델을 선정함
  - ✓ 2주간 수집한 '정상' 데이터를 사용하여 선정된 시계열 파운데이션 모델을 Fine-Tuning
  - ✓ 모델은 PC 시스템 자원의 정상 패턴을 깊게 학습하여, 여기서 벗어나는 패턴을 '이상'으로 간주함
- **이상 점수 정의**
  - ✓ **예측 오차 기반 접근:** 모델이 학습한 정상 패턴을 바탕으로 한 예측값과 실제 관측값 간의 차이를 이상 점수로 사용

$$\text{Anomaly Score}(t) = \text{Error}(\text{Predicted}(t), \text{Observed}(t))$$

- **성능 검증**
  - ✓ Precision, Recall, F1-Score 와 같은 지표를 사용하여 모델을 정량적으로 평가
- **모델 비교 및 개선**
  - ✓ Reconstruction 기반 모델, Forcasting 기반 모델의 성능을 비교하고 PC 시스템 이상탐지에 가장 적합한 모델 선정
  - ✓ 과잉 일반화 문제 등 모델 차원의 문제 발생 시 모델 구조 개선 방안을 탐구할 계획이 있음

Anomaly Scenario

Use Case 1



작업물 손실

메모리 과부하로 프로그램이 중단되어
작업중이던 자료가 날아간 상황 (ex. PPT)

Scenario 1

점진적 메모리 누수

특정 프로세스의 버그로 인해 메모리 사용량이
눈에 띄지 않게 아주 조금씩
지속적으로 증가하는 상황을 재현

Contextual Anomaly
탐지 여부 검증

Use Case 2



주기적 백업 실패

주기적으로 작동해야 하는 사진 백업
프로그램이 특정 문제로 작동되지 않음

Scenario 2

주기적 패턴 소멸

정각에 실행되던 백업 스크립트가 실패하여
평소 주기적으로 나타나던 CPU 및 디스크
사용량의 스파이크가 사라지는 상황을 재현

시간적 패턴 자체
학습 여부 검증

Use Case 3



보안 위협

해커의 공격으로 인해 PC 내부의 데이터가
네트워크를 통해 대량으로 유출됨

Scenario 3

비정상적 상관관계 패턴

악성코드 감염 등으로 인해 CPU 사용량 변화
없이 네트워크 트래픽만 폭주하는 상황을 재현

지표 간의 상관관계 이상 발생 시
탐지 여부 검증

Demonstration

- InfluxDB 데모 시연

Thank you

Thank you for listening to the presentation

Pohang University of Science and Technology

Data Engneering Team 7